

Beyond the Scrollbar: An Evolution and Evaluation of Alternative List Navigation Techniques

D. Scott McCrickard

College of Computing and
Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332-0280
+1 404 894 9390
mccricks@cc.gatech.edu

Richard Catrambone

School of Psychology and
Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332-0170
+1 404 894 2682
rc7@prism.gatech.edu

ABSTRACT

List navigation devices such as scrollbars provide a simple way to browse and find items in a list, but they typically do not communicate information about individual list entries. This paper presents two alternative devices, a mural bar and a pile bar, which encode information about each entry into the trough of the bar. Results from an experiment suggest that these new devices lead to improved user performance on typical list-based tasks.

Keywords

Scrollbar, information mural, pile metaphor, navigation bar, evaluation

INTRODUCTION

Many computer applications require a user to select items from a *list*, a series of related items that is often displayed using a rectangular widget called a *listbox*. For example, email programs require a user to select messages listed by sender or subject, and file browsers require a user to select files listed by filename. A user can select items by clicking on them using a pointing device such as a mouse. However, since lists are often too long to be displayed in their entirety in the available space in the listbox, only a subset of items can be seen at any given time. Some sort of navigation device must be provided to help the user traverse the list and find items of interest.

One common navigation device is the *scrollbar*, a simple widget that actually provides a great deal of power and information without requiring much space. A scrollbar provides the ability to change the portion of the list that is visible in the listbox while constantly displaying information about the relative size and position of the visible portion. Scrollbars are not only used with listboxes, but with any textual or graphical entity that cannot be displayed in its entirety in the

available space, including large images and long documents. While this flexibility is important, if a software designer is interested in displaying information about a list, the flexibility gained might be at the sacrifice of information about the list that is communicated. It would be advantageous to maintain the scrollbar/listbox navigation paradigm while increasing the amount of information that can be provided.

This would allow current computer users to build on an existing mental model and hopefully shorten the time required to learn the interface. In addition, in many applications scrollbars could be replaced with the improved navigation devices without major changes to layout or functionality.

One method for communicating more information is to use the space inside the scrollbar to represent each list entry with a graphical line, where properties of the list entries are reflected in properties of the graphical lines. For example, the length of the list entry could be represented by the length of the line, and the position of the entry in the list could correspond to the position of the line in the scrollbar area. Categories or classifications of the entries could be shown using the color, size, or offset of the line.

A number of systems use this type of graphical representation to communicate information. Data visualization sliders provide a scrollbar-like interface for specifying numeric values [2]. The space inside the sliders contains a color-coded barplot or density plot representing the range of numeric values. Fisheye views, originally described in [3], have been adapted to text interface devices in systems such as [5]. A fisheye view maintains context by displaying the entire information space while focusing on an enlarged portion of it. Value bars represent a list entry using thin colored stripes next to the scrollbar [1]. The color and size of a stripe represents the goodness of the attribute for a particular entry. The attribute-mapped scrollbars of read wear and edit wear map data about editing changes to marks on a scrollbar [6].

While these systems provide alternatives to scrollbars for certain tasks, few experiments have been conducted to determine whether these alternatives actually improve performance for typical scrollbar tasks. This paper describes two navigation bars that use the line-based list representation: the mural bar and the pile bar. The *mural bar* uses an information mural [4] to communicate list entry properties, while the *pile bar* employs a pile metaphor [9] to arrange the encodings. We expect that these navigation bars will provide more information than a scrollbar about the contents of the list in a comparable amount of space, thus assisting the user in certain list browsing and accessing tasks. We examine this hypothesis by describing a number of scrollbar-based tasks and testing whether users can perform these tasks more quickly and easily with a mural bar or pile bar than with a scrollbar.

LIST NAVIGATION TECHNIQUES

A common way to navigate through a list is with a scrollbar. Arrows at the top and bottom of the scrollbar can be clicked and held with the mouse button to change the visible portion of the list. A rectangular bar, called the *thumb*, provides information about the relative size and position of the listbox entries as well as another means to navigate within it. The thumb slides up and down within a *trough* as the list is scrolled up and down. The size and position of the thumb in the trough are proportional to the size and position of the visible portion of the list. Clicking in the trough above or below the thumb jumps the display toward the clicked location.

As we discussed in the introduction, scrollbars are very flexible in that they can be used with a variety of different devices. However this flexibility limits the amount of information conveyed

by the scrollbar. To address this problem, we created two alternative navigation bars: the mural bar and the pile bar. The mural bar and pile bar contain many of the features and functionality of a scrollbar, plus they include graphical encodings in the trough area that display information about the items in the list. They have arrows above and below the encodings for scrolling through the list. The thumb surrounds the encodings whose elements are shown in the textual list. The user can drag the thumb or click on the encodings to shift the visible portion of the list.

In both the mural and pile techniques, a list item is represented as a colored horizontal line in a vertically-oriented bar. The lines are in the same order as the corresponding list entries. The length of each line is proportional to the length of its list entry, so a longer entry would be represented by a longer line. Certain lines are colored to correspond to groupings of list items. For example, a listing of files might be colored by file type (red could indicate html files, blue could indicate Java code files, and so on) or a listing of email messages might be colored by sender name. Repeated color patterns in the encodings can then reveal related items. Thus, the user could quickly detect related items by their similar encodings. Entries that do not belong to a specific category might be represented by grey lines.

While the mural and pile techniques are similar in their representation of individual entries, they differ in their allocation of the available screen real estate to these representations. The information mural technique shrinks the list representations into the available space. Thus, the larger the list, the less space each item receives for its representation. In contrast, the pile view technique provides each list representation with a fixed amount of space, then allows the representations to overlap if necessary. This technique requires extra information on stacking order: if two or more entries overlap, one of them must be drawn on top. The easiest approach (and the one taken in our experiment) is to stack highlighted representations on top so they are most visible. However, if there are a large number of highlighted entries, certain entries can be completely obscured.

Mural Bars

Information murals, compressed graphical displays of an information space, have been used in a wide range of applications, including visualizations of software execution, numerical data, and general information [4]. Information murals represent a large information space (in this case a textual list) with a smaller graphical space by mapping the elements in the large space into the smaller one. In the case of a list, the characters in the list map to the pixels in the navigation bar to create a scaled down picture of the entire list. To understand how the information mural is created, think of a rectangle that is M pixels wide and N pixels high as an $M \times N$ grid that overlaps the entire list. A pixel in the grid is shaded with intensity corresponding to the number of characters that touch it: the more characters that map to a pixel, the darker it becomes. The shading effect that results from this mapping shows the size and number of elements in the list.

The mural bar looks and acts much like a scrollbar except it has an information mural in the trough area (see Figure 1a). The thumb surrounds the entry representations that are visible in the listbox. As with a scrollbar, clicking above or below the thumb (where the other representations lie) will advance the listbox display toward the clicked spot.

Information murals provide an overview of the entire list using whatever space is available. Since an equal amount of space is given to each item, any item can be seen with equal alacrity no matter where they are in the list. Murals are particularly effective at highlighting a series of related items with identically colored representations since the representations run together to create a larger and more noticeable colored block. However, information murals can obscure

details about individual elements in the list. Larger lists devote fewer pixels to any single entry, making it more difficult to find individual entries. If certain entries are known to be important, a method is needed for highlighting them. We will see such a method in the pile bar.

Pile Bars

The pile metaphor introduces onto the desktop the concept of piling items on top of each other. Just as documents can be piled on a desk and identified by their appearance, items on the computer desktop can overlap and be recognized by their color, size, or shape. The pile metaphor was introduced in a desktop document layout system [9]. Piles created by the user had a disheveled appearance with parts of the document icon sticking out slightly so the user could find a document even if it were in the middle of the pile.

The pile bar employs the pile metaphor in the arrangement of graphical bars. In the pile bar, each entry is represented by a bar with a fixed height of five pixels (see Figure 1b). To show all of the representations at this fixed size, the entry bars must be piled on top of each other, meaning that some bars may be partially or completely obscured. Each bar is drawn with a slight random horizontal offset, making it possible to see bars not on the top of the pile. An additional protocol is needed to describe the stacking order of the pile. For example, the last access time could be used in a protocol, with the most recently selected items appearing on top of the pile. In our experiment, we place the colored entries on top of the pile.

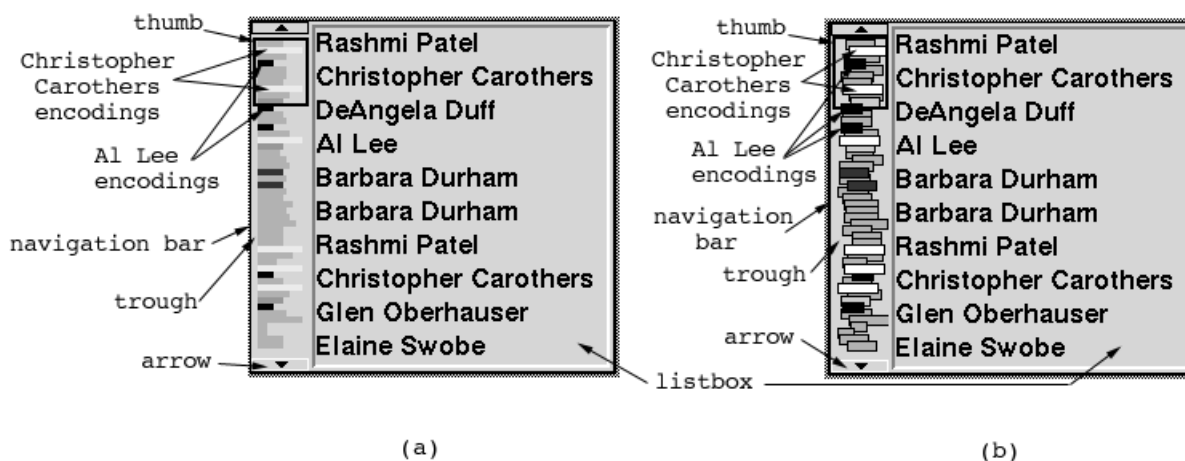


Figure 1. An mural bar (a) and pile bar (b) with their accompanying listboxes. For both bars, each list entry is represented with a horizontal line. The thumb surrounds the representations that are shown in the listbox. The user can click on the arrows or in the trough to adjust the position of the list.

The pile bar would be useful in finding items of the same type as a selected one. For example, with lists of email messages colored by sender name, it would be easy to find prior items from the selected person as they would rise to the top of the stack. Piles would also assist in identifying recently accessed messages.

Since this technique relies on additional heuristics for the ordering of colored bars, a poor heuristic may lead to obscured bars for desired items such that the user is unable to find items of interest. Also since this technique relies on the random offset to ensure that more items are visible, structured lists such as directory listings, where the offset of lines has meaning, would not work as well with this display.

IMPLEMENTATION AND USE OF NAVIGATION BARS

To test our ideas we implemented a navigation bar widget as an extension to the Tcl/Tk toolkit [8]. The widget can display either an information mural or a pile view in a scrollbar trough. The command set of a navigation bar is a superset of the commands for a Tcl/Tk scrollbar, so programmers can use navigation bars in situations where scrollbars typically would be employed. Of course, the navigation bar widget contains additional commands that provide control over the color and layout of the bars.

Navigation bars have been incorporated in a number of programs as a replacement for scrollbars. One such program is *exmh*, a graphical email reader [10]. Though *exmh* supports filtering and nested folders, lists of email messages can still become large and difficult to navigate. The navigation bar can highlight messages by sender name. A user can provide a list of names to be highlighted and the colors to use, and the navigation bar will highlight all messages sent by these people. This makes it easier to find messages from a particular person.

Navigation bars were also added to *Irwin*, an information monitoring and display system [7]. *Irwin* is often used to monitor wire news articles available on the World Wide Web and create clusters of related articles. The navigation bar is used to highlight the most closely related clusters. If the pile view is used, the user can provide keywords that will identify the most interesting articles then raise them to the top of the pile.

Navigation bars have also been used with other smaller programs such as file selectors and object-oriented class browsers. While we expect that navigation bars can and will be used with other types of programs as well, it is still necessary to show that they can produce a reliable advantage for certain tasks over tools using scrollbars.

EXPERIMENT: EVALUATING THE NAVIGATION BARS

Both the mural bar and the pile bar provide more information about the contents of the list than does a scrollbar. The nature and functionality of the pile and mural bars allows a user to access this information using techniques similar to those used with a scrollbar. We expect that the additional information and familiar access techniques will allow a user to access information more quickly and easily than with a standard scrollbar without the additional learning time required by more complex devices. In addition, one issue of interest is how these two approaches will work relative to the scrollbar on larger lists. That is, while we predict advantages on smaller lists for the two graphical encoding techniques, it is unclear what will happen on larger lists. Will the two approaches increase their expected advantages over the scrollbar or will the advantages disappear as the graphical displays become more cluttered? We are also interested in whether performance with the graphical techniques improves over trials as the users get more comfortable with them and possibly learn to take advantage of their features.

Our plan is to use colorings to create meaningful patterns that will convey information to the user about the list contents, thus lessening the need to scroll through a textual list of document headers. Users should be able to associate graphical patterns with their documents, and repeated patterns will indicate related list items.

Based on our assumptions about how users would take advantage of the navigation techniques, we made a number of predictions about how the encodings would lead to better performance. In our experiment, we measured the time required for participants to perform various list-based tasks using one of the navigation bars: mural bar, pile bar, or a typical scrollbar. A given participant used only one of the three types of navigation bars. By having each participant

practice and use a single type, we hoped to advance the expertise of each participant to the point where the technique employed could be used quickly and efficiently.

In our experiment, we chose email sender names as the items in the list. Participants made estimates about the size and contents of eight different lists and performed several commonly occurring search tasks on each list.

Our goal was to determine whether the navigation bars described in this document help users understand, examine, and access information in a list more quickly and easily than with a scrollbar. The remainder of this section outlines the rationale used in constructing the lists and the tasks, and describes the experiment.

Selecting the Lists

We chose to have users perform tasks on lists of names. The names were taken from entries in an email mailing list, with slight changes and additions made to support user tasks. A scrollable list of names occurs frequently in applications; for example, many email and news readers provide a list of messages showing the sender names. This type of list can have repeated items in the list and clusters of items with the same name, either of which can be highlighted by the navigation bars. Since these types of lists grow and change often, the graphical representations and highlights should help users discover the size and content of the list and find items more quickly.

The number of items in a list varied in order to allow us to examine whether performance differences (if any) due to type of navigation bar used are constant across list lengths or whether they are affected by list length.

Selecting the Tasks

In selecting tasks for this experiment, we wanted to choose tasks that exemplified the types of tasks performed with scrollbars. We realize that these tasks do not encompass all possible tasks that are performed with a scrollbar.

We hoped participants would be able to deduce characteristics about the list based on the appearance of the navigation bar. The more information that users can determine about the size and contents of the list, the better off they will be when they must navigate through the list and find items. We tested the utility of the navigation bars by asking each participant to estimate, as quickly as possible, the total number of entries in a list and the number of entries with some label (such as the number of entries with a particular person's name). The list lengths were large enough so that they could not be quickly counted. The number of labeled entries to be estimated were typically about 15-25 percent of the total number of entries. Again, we expected that participants would not be able to quickly count the number of labeled entries but would have to estimate them using information from the navigation bar and/or from the listbox.

The next task for each list required participants to find and select items from the list. In one task, participants had to find and select three items with a particular label. In another, participants had to find a cluster of items with a particular label and select one. These items to be selected were highlighted in the navigation bars (for those participants using the pile bar or the mural bar). We expected these highlights to help participants to find the items more quickly. Finding items with some label is a typical task that requires the user to scroll through the list, scanning the text until the item is found. We thought that finding individual items would be easier with a pile bar, since the items would be on top of the pile and would have a larger number of pixels than with the mural bar. On the other hand, finding a cluster of items should be easier with the mural bar since the representations merge to form a larger target while in the pile bar the representations overlap.

Method

Participants

Seventy-six Georgia Tech undergraduates participated in this experiment for class credit. All participants had 20/20 (corrected) vision and none were colorblind.

Apparatus

The experiment was conducted on Sun Sparcstation 2 workstations connected to a 15-inch color monitor. To perform the tasks, participants used the Sun optical mouse.

Movement Recording

The time and position of all mouse clicks on the interface were recorded. The result of the click was noted as well; that is, whether the click caused a list entry to be selected or caused the navigation bar to scroll up or down one line or to scroll up or down one page. The time at which each task was successfully completed was noted as well.

Design

The independent variable was type of navigation bar used, thus there were three groups in the experiment: Information Mural ($n = 25$), Pile View ($n = 31$), and Scroll Bar ($n = 20$). The unequal group sizes were due to unanticipated scheduling problems. Dependent variables were

time to do each task, the accuracy of each estimation task, and the technique used for each task (described below).

Procedure

Participants were run in small groups, one participant per computer. Participants first completed a questionnaire in which they provided background information such as their major, GPA, an estimate of the number of hours per week they spent on the computer, and the types of applications they typically used so that we could examine whether these factors were related to performance. The on-line portion of the experiment began with a series of introductory screens that described the experiment and gave participants warm-up tasks to ensure they could use the mouse properly to select list items.

Following the instructions and warm-up, participants began the experimental tasks. There were four tasks for each of eight lists. The lists varied in size from 34 items to 195 items (see Table 1 for details about the lists). The tasks to be performed on each list were, in order: 1) Estimate the size of the list, 2) Estimate the number of items from the list with a particular label (e.g., Al Lee), 3) Find three items from the list with a particular label (e.g., three messages from Christopher Carothers), and 4) Find a cluster of three items (i.e., three consecutive items) from the list with a particular label (e.g., three messages from Al Lee).

List	List Size for Task 1	Number of Items with a Particular Label for Task 2	Location of Items to be Selected for Task 3	Location of Cluster of Items to be Selected for Task 4
1	45	9	5,19,29	36,37,38
2	105	20	25,52,75	90,91,92
3	60	12	16,30,38	46,47,48
4	52	15	18,34,46	27,28,29
5	85	12	15,38,63	75,76,77
6	34	17	5,24,30	20,21,22
7	195	22	33,79,154	21,22,23
8	73	10	16,37,63	54,55,56

Table 1. Solutions for the tasks for each list. Task 1 asked for a list size estimate. Task 2 asked for an estimate on the number of labeled items. Task 3 asked that three highlighted items be selected (their position in the list is given above). Task 4 asked that one of a cluster of items be selected (the positions of each item in the cluster is given above).

For each task, the participant first read a description of the task and then, when the participant was ready to begin the task, he or she would click the "Begin Timed Task" button which would cause the listbox and navigation bar to be displayed and would also start the timer. The timer stopped as soon as the task was completed. At that time, the 'Next' button was activated and the participant could click on it in order to move on to the next task.

The order of the tasks was the same for all groups, the only difference was the type of interface used to do the tasks (scrollbar, pile bar, or mural bar). For the two tasks on each list in which participants made estimates (Tasks 1 and 2), it was stressed that a good time with some error was better than a poor time with little or no error. Participants were not given feedback on the accuracy of their estimates during the experiment. For the two search tasks (Tasks 3 and 4), the entries to be selected were spaced so that the participant had to use the navigation bar to find them (they were not immediately visible in the listbox). The entries to be found were highlighted with some color in the navigation bar (for participants using the pile bar or mural bar).

Expected Results

The performance of the participants for each task was expected to depend on the length of the list and the type of encoding used in the navigation bar. Longer lists result in more cluttered displays and greater distances to traverse between search tasks. People are particularly good at identifying unique patterns in graphical displays, but it seems that increasing the number of items should increase the time required to find and select items. As we analyze how the participants will perform the tasks, we will consider how the size and encoding type affect the ability to process the information necessary to complete the task.

Consider the task of estimating the size of a list. Using a scrollbar, one approach might be to estimate the number of entries visible in the listbox, then estimate the ratio of the thumb size to the trough size, then calculate the total size. For example, if there are five entries in the listbox and the thumb occupies about a fourth of the trough, then there must be about 20 entries in the entire list. For the information mural and pile view, the ratio of the number of entries in the bounding box to the total number of entries will provide similar information. However, in addition they show a representation for each entry in the list (with the proviso that in the information mural the entries can become quite small as the list gets large and that in the pile view entries can be obscured by other entries that fall on top of them). Nevertheless, both the information mural and pile view should show better performance compared to the scrollbar because both techniques provide two methods for judging list size compared to only one for the scrollbar.

For the task of estimating the number of entries of some type, the scrollbar will present problems because it does not contain representations of the individual items. Thus, the user must scroll through the list and observe the contents of the listbox. The information mural and pile view should provide faster estimates (with little or no loss of accuracy) since the color codings mean that no scrolling is required.

For the two search tasks, the pile view and information mural should show better performance than the scrollbar as the encodings are always visible (though smaller or partially obscured) in these bars. We expect the pile view will produce better performance than the information mural when a user has to find individual items (Task 3) because the graphical representations will be larger, making them easier to see and select. However, when finding clusters of items, the information mural will be better because the multiple items of the same type will create a larger representation, while in the pile view the representations will overlap without taking up much more space. We expect the scrollbar to have the worst performance on all search tasks since users must scan the text to find items and they must scroll the lists to see the representations.

RESULTS

The average number of correct Task 1 (estimating size of list) and Task 2 (estimating number of list items with a certain label) estimates for each group across the eight lists are presented in Table 2. For data analysis purposes, we considered an estimate to be correct if it was within 15 percent of the actual value (see Table 1 for the correct answers to each task). Table 2 also shows the average times for doing these tasks.

There was no significant difference among the three conditions on the accuracy of estimating list lengths (Task 1), $F(2,73) = 1.40$, $MSE = 4.54$, $p = .25$ or in estimating the number of list items with a certain label (Task 2), $F(2,73) = 0.78$, $MSE = 3.92$, $p = .46$. There was also no difference among the groups in the time to do Task 1, $F(2,70) = 1.62$, $MSE = 47.61$, $p = .21$ (this average does not include the time for estimating the size of list #7 since that list was considerably longer than the other lists). However, there was a difference among the groups in the time for estimating the number of list items with a certain label, $F(2,73) = 5.78$, $MSE = 79.42$, $p = .005$. Pairwise comparisons indicate that both the Information Mural (IM) and Pile View (PV) conditions were faster than the Scroll Bar (SB) condition while the first two groups did not differ.

The average times for doing Task 3 (finding three items in the list with a certain label) and Task 4 (finding a cluster of three items with a certain label) are shown in Table 3. There was a significant difference among the groups in the time for finding three list items with a certain label, $F(2,73) = 16.59$, $MSE = 59.17$, $p < .0001$. Pairwise comparisons indicate that both the IM and PV conditions were faster than the SB condition while the first two groups did not differ. Similarly, there was a significant difference among the groups in the time for finding a cluster of three list items with a certain label, $F(2,73) = 4.11$, $MSE = 10.17$, $p = .02$. Pairwise comparisons indicate that both the IM and PV conditions were faster than the SB condition while the first two groups did not differ.

Group	Task 1 Corr. (max.=8)	Time for Task 1 (sec.)	Task 2 Corr. (max.=8)	Time for Task 2 (sec.)
Info. Mural (IM)	5.36	17.72	5.40	18.98
Pile View (PV)	5.03	15.95	5.90	15.09
Scroll Bar (SB)	6.05	19.51	6.10	23.76

Table 2. Average number of correct estimates and times for those estimates for Tasks 1 and 2 as a function of condition.

Group	Time for Task 3 (sec.)	Time for Task 4 (sec.)
Information Mural (IM)	18.58	7.14
Pile View (PV)	17.60	6.61
Scroll Bar (SB)	29.54	9.18

Table 3. Average times to find list items with certain labels (Task 3) and to find a cluster of list items with a certain label (Task 4).

Finally, Table 4 shows the frequency, collapsed across all tasks, with which participants used various methods for moving through the lists in order to gather information for making estimates or for finding specific items. For some tasks, participants were able to identify a solution by simply looking at the navigation bar. We inferred that this "Look" technique was being used if there were no mouse clicks on the navigation bar prior to a participant typing in their estimate or selecting an item in the listbox. For other tasks participants chose, or had, to page or scroll (or both).

There was a significant difference among the groups in the frequency of using the different techniques when doing different tasks. Consider Task 1 (estimate the size of a list) in Table 4. In principle participants in all three conditions could have used the Look technique, that is, they could have determined the number of items in the listbox and then multiplied that value by the number of times the thumb would have to be moved in order to go to the bottom of the trough. In fact though, participants in all conditions used a variety of techniques although they tended to use the item by item scrolling the least often. The groups did not differ from each other in the frequencies of use of each technique.

Task 2 required participants to estimate the number of items with a certain label. Once again, the IM and PV conditions could do this task by observing information in the navigation bar; however, the SB condition could not. This time the IM and PV groups used the Look technique more often than the SB condition, $F(2,73) = 9.63$, $MSE = 7.33$, $p = .0002$. Conversely, participants in the SB condition tended to use the other techniques more often than the other conditions although only the difference in use of the Scroll + Page technique is statistically significant, $F(2,73) = 9.42$, $MSE = 1.50$, $p = .0002$.

Tasks 3 and 4 required participants to find items with the same label. These tasks required participants to click on items in the navigation bar, thus, by default, no one could use the Look technique. Interestingly, we found that IM and PV participants were more likely than SB participants to move through the lists by taking larger steps as shown by the former two groups' greater likelihood of using the Page technique for both Tasks 3 and 4 compared to SB participants ($F(2,73) = 13.20$, $MSE = 6.00$, $p < .0001$; $F(2,73) = 3.83$, $MSE = 5.32$, $p = .03$, for Tasks 3 and 4, respectively; see Table 4). Conversely, SB participants were more likely than IM and PV participants to use the slower item by item scroll technique for both tasks ($F(2,73) = 3.65$, $MSE = 4.53$, $p = .03$; $F(2,73) = 4.34$, $MSE = 4.31$, $p = .02$, for Tasks 3 and 4, respectively; see Table 4).

Group	TASK 1 (Estimate List Size)				
	Look	Page Up or Scroll Item by	Page Down	Item by	Scroll + Page
Information Mural (IM)	1.56	3.24	0.36		2.84
Pile View (PV)	2.61	2.39	0.97		2.03
Scroll Bar (SB)	2.60	2.25	0.65		2.50

TASK 2 (Estimate Items with Certain Label)

Information Mural (IM)	4.16	1.92	1.16	0.76
Pile View (PV)	5.36	1.71	0.71	0.23
Scroll Bar (SB)	1.95	2.25	2.05	1.75

TASK 3 (Find Three Items)

Information Mural (IM)	0	6.40	0.76	0.84
Pile View (PV)	0	6.45	0.84	0.71
Scroll Bar (SB)	0	3.15	2.30	2.55

TASK 4 (Find Cluster of Items)

Information Mural (IM)	0	7.04	0.60	0.36
Pile View (PV)	0	7.16	0.61	0.23
Scroll Bar (SB)	0	5.45	2.20	0.35

Table 4. Average number of times participants used various techniques for moving through lists in order to do the various tasks. "Look" means the user did not move through the list (other than presumably to scan it visually). "Page Up or Down" means the user clicked above or below the thumb in order to move the list up or down one thumb's-worth at a time. "Scroll Item by Item" means the user clicked on one of the arrows at the top or bottom of the trough to move the list up or down one item at a time. "Scroll + Page" means the user utilized both the item and page techniques to move the list up or down.

The tendency of the IM and PV participants to use more efficient techniques for finding information in a list is consistent with the finding that they did most of the tasks more quickly than SB participants. A regression analysis was carried out on the time to do the various tasks as a function of various predictors such as type of navigation bar used, technique used for that task, and other factors (details are available from the authors). The summary finding was that technique was typically a strong predictor of task time. In addition, technique interacted with list length suggesting that the advantage of an efficient technique grew stronger for longer lists.

DISCUSSION

The estimation accuracy and times showed equal accuracy among the different interfaces but a tendency towards faster performance for the mural and pile bar groups relative to the scrollbar group (at least for one of the estimation tasks). The search results clearly indicate faster performance by the mural and pile bar groups compared to the scrollbar group. Performance by the mural and pile bar groups did not reliably differ from each other. Taken as a whole these results suggest that graphical interfaces that attempt to encode certain features of list entries can improve users' accuracy and speed as they do typical tasks such as trying to get a sense of the number of related items in a list or the location of certain related items.

The search technique results suggest that the mural and pile bar interfaces encouraged users to move through the list more efficiently. Obviously on some tasks (i.e., estimating list length and number of entries with a particular label) there was no need to physically move through the list since all entries were graphically represented in the trough. However, for tasks that did require moving through the list, mural and pile bar users were more likely to move in larger units using the Page approach while scrollbar users were more likely to move item by item. This may be because mural and pile bar users could see the graphical representations of the items they wished to find and thus, could take larger leaps towards those items. Scrollbar users, on the other hand, had to move more "blindly" through the lists searching for the desired items. Thus, they may have been inclined to be more cautious. This difference in efficiency grew larger as the list length increased. This suggests that the benefits of the graphical encodings used in the information mural and pile view were not eliminated by increased clutter as the lists grew longer.

CONCLUSIONS

The results of our experiment are promising for the use of navigation bars in new interfaces. We showed the advantages of graphical representations for lists with repeated or related entries, and the new landscape of online information provides many instances of such lists: email, Usenet news, online news articles, and various other lists on the World Wide Web. While filtering the information is important, the navigation bar provides an additional way to specify and locate potential items of interest.

In the future we plan to conduct further experiments to determine whether and when the pile view and information mural show better performance. We expect that finding items in larger lists with fewer highlighted items will show better performance with the pile view. On the other hand, the information mural should show better performance with denser clusters of items since they appear larger in the mural while they merely overlap in the pile. We also plan to vary certain parameters of the views such as position, order, and size to create other visual paradigms. Finally, we hope to identify typical qualities of lists from various sources to see if we can develop recommendations for which types of views promote better user performance.

REFERENCES

1. Chimera, R. Value Bars: An information visualization and navigation tool for multi-attribute listings. In *Proceedings of the ACM Human Factors in Computing Systems Conference (CHI '92)*. Monterey, CA, May 1992, 293-294.
2. Eick, S.G. Data Visualization Sliders. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '94)*. November 1994, 119-120.
3. Furnas, G.W. Generalized Fisheye Views. In *Proceedings of the ACM Human Factors in Computing Systems Conference (CHI '86)*. Boston, MA, April 1986, 16-23.
4. Jerding, D.F., and Stasko, J.T. The Information Mural: A Technique for Displaying and Navigating Large Information Spaces. In *Proceedings of the IEEE Symposium on Information Visualization*. Atlanta, GA, October 1995, 43-50.
5. Greenberg, S. A fisheye text editor for relaxed-wysiwiw groupware. In *Companion for the ACM Human Factors in Computing Systems Conference (CHI '96)*, Vancouver, BC, Canada, April 1996, 206-207.

6. Hill, W.C., and Hollan, J.D. History-enriched digital objects: Prototypes and policy issues. *The Information Society*, 10 (2), April 1994, 139-145.
7. McCrickard, D.S., and Rowan, T.H. Monitoring and visualizing information resources. *Technical Report ORNL/TM-13193*. Oak Ridge National Laboratory, July 1996.
8. Ousterhout, J.K. *Tcl and the Tk Toolkit*. Addison-Wesley Publishing Company. Reading, MA, 1994.
9. Rose, D.E., Mander, R., Oren, T., Ponceleon, D.B., Salomon, G., and Wong, Y.Y. Content Awareness in a File System Interface: Implementing the 'Pile' Metaphor for Organizing Information. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR '93)*. Pittsburgh, PA, June 1993, 260-269.
10. Welch, B.B. Flexibility and Customization in the EXMH mail user interface. In *Proceedings of the USENIX Third Annual Tcl/Tk Workshop*. Toronto, Canada, July 1995, 261-267.